Analysis and Evaluation of Sketch Recognizers in the Creation of Physics Simulations

Abílio Costa, School of Engineering (ISEP), Polytechnic of Porto, Porto, Portugal
João P. Pereira, School of Engineering (ISEP), Polytechnic of Porto, Porto, Portugal

ABSTRACT

Sketch-based interfaces can provide a natural way for users to interact with applications. Since the core of a sketch-based interface is the gesture recognizer, there is a need to correctly evaluate various recognizers before choosing one. In this paper the authors present an evaluation of three gesture recognizers: Rubine’s recognizer, CALI and the $1 Recognizer. The evaluation relied on a set of real gesture samples drawn by 32 subjects, with a gesture repertoire arranged for use in SketchyDynamics, a programming library that intends to facilitate the creation of applications by rapidly providing them with a sketch-based interface and physics simulation capabilities. The authors also discuss some improvements to the recognizers’ implementation that helped achieving higher recognition rates. In the end, CALI had the best recognition rate with 94% accuracy, followed by $1 Recognizer with 87% and finally by Rubine’s recognizer with 79%.

Keywords: CALI, $1 Recognizer, Gesture Recognition, Physics Simulation, Rigid-Body Dynamics, Rubine, Sketch-Based Interfaces

INTRODUCTION

Using pen and paper to draw or sketch something in order to express an idea is very common and also very natural for us. By using this concept in user interfaces one can make the interaction process more natural and spontaneous.

With this concept in mind, we developed a programming library that aims to aid in the creation of applications for two-dimensional physics simulations in which the user interacts directly with the scene using a “pen and paper” style interaction. Thus, instead of selecting from a menu which objects compose the scene to be simulated, the user can simply draw the objects directly in the scene. We hope that developing a library that integrates a calligraphic interface with a physics simulation engine will encourage developers to create new applications around this concept, be them for educational purposes, like an application used to teach physics to students using an interactive whiteboard, or for entertainment purposes, such as a physics-based game where the user draws parts of the scene in order to reach a goal, in the same genre as Crayon Physics Deluxe (Purho, 2009). These are only two examples of a wide range of possibilities.

The library will support three gestures to draw primitives and other three to define relationships between primitives. The first three

DOI: 10.4018/ijcicg.2013070101
gestures instantiate rectangles, triangles and circles, which can be created by drawing these symbols directly. To establish relationships between primitives the user can draw a zigzag to connect two primitives with a spring, a cross to pin a primitive over another and a small circle to connect one primitive over another with a rotation axis. Since both the circle primitive and the rotation axis relationship use the same gesture and the distinction is made by the system based on the gesture’s size and context, we only have in fact five gestures to recognize, presented in Figure 1. Given that the cross is the only gesture that cannot be drawn with only one stroke, we opted to replace it with an alpha, which is an intuitive single-stroke representation of a cross. We chose to use only single-stroke gestures because, besides meeting the needs of our library, it makes the interaction simpler.

Since the user must interact in a way that is as natural and unrestricted as drawing with a pen on a paper, we conducted an evaluation of various gesture recognizers in order to select the one that best fits our needs. In this evaluation we have organized two sessions to collect samples of each gesture, drawn by various subjects, in order to put the recognizers to test with a wide range of data. This paper describes that evaluation in detail, along with various considerations to achieve higher recognition rates. To further contextualize this evaluation, we also describe the programming library in which the selected recognizer was integrated.

In the next section we present an overview of the related work done in the gesture recognition field. We then present the preparation of the recognizers’ evaluation, how it was conducted and its results. A brief description of the developed programming library and its functionality is then presented. Finally we propose potential future developments of this work and present our conclusions.

RELATED WORK

Given the potential of automatic sketch recognition, a lot of work has been done in order to develop recognizers capable of dealing with the intrinsic ambiguity in hand-drawn sketches. Since there are a great variety of sketch recognition algorithms, it is only natural that there’s also diversity in their characteristics. For example, some recognizers only work with single-stroke sketches, while others are oriented towards multi-stroke. Also, whether or not the recognizer can identify sketches independently of their orientation, scale, and drawing order can greatly affect its usefulness in some domains. Another important characteristic is if the recognizer can be trained with new gestures, meaning that it can be easily expanded, or rather if its gestures are hardcoded, which makes it difficult to change its gesture set to fit a new domain.

Rubine’s recognizer (Rubine, 1991), a trainable gesture recognizer, classifies each gesture using a linear classifier algorithm with

Figure 1. Set of gestures used in our work

![Figure 1. Set of gestures used in our work](image-url)
Analysis and Evaluation of Sketch Recognizers in the Creation of Physics Simulations